Manual Tecnico – JavaBridge

Estados del Scanner

Autómata Finito Determinista (AFD)

Estado S0 – Estado Inicial

Reconocimiento inicial de todos los caracteres del código fuente.

Transiciones:

- Caracteres alfabéticos (A-Za-z) => S1 (Identificadores/Palabras reservadas)
- “ => S2 (Cadenas de texto)
- ‘ => S4 (Caracteres)
- dígitos (0-9) => S7 (números enteros/flotantes)
- > => S10 (Operadores relacionales)
- < => S13 (Operadores relacionales)
- = => S15 (Asignación/Comparación)
- ¡ => S17 (Diferencia)
- + => S19 (Suma/Incremento)
- - => S21 (Resta/Decremento)
- * => S23 (Multiplicación)
- / => S24 (División/Comentarios)
- (=> S 29 (Parentesis izquierdo)
-) => S30 (Parentesis derecho)
- { => S31 (Llave izquierda)
- } => S32 (Llave derecha)
- ; => S33 (Punto y coma)
- , => S34 (Coma)
- [=> S35 (Corchete izquierdo)
-] => S36 (Corchete derecho)
- . => S37 (Punto)

Estado S1 – Reconocimiento de identificadores

Estados para identificadores y palabras reservadas

- [A-Za-z0-9] => Permanece en S1
- Otro carácter => Retorna token TK_id o paralabra reservada

Estados para Cadenas y Caracteres

S2 – Inicio de cadena

- [^”\n] => Permanece en S2
- ” => S3

S3 – Fin de cadena => Retorna TK_str

S4 – Inicio de carácter

- [^’\n] => S5

S5 – Contenido de carácter

- ‘ => S6

S6 – Fin de carácter => Retorna TK_char

Estados para Numeros

S7 – Enteros

- [0-9] => Permanece en S7
- . => S8
- Otro => Retorna TK_init

S8 – Punto decimal

- [0-9] => S9

S9 – Parte decimal

- [0-9] => Permanece en S9
- Otro => Retorna TK_float

Estados para Operadores

S10 - >

- > => S11 (>>)
- = => S12 (>=)
- Otro => Retorna TK_greater

S13 - <

- = => S14 (<=)
- Otro => Retorna TK_less

S15 - =

- = => S16 (==)
- Otro => Retorna TK_assign

S17 - ¡

- = => S18(¡=)

S19- +

- + => S20(++)
- Otro => Retorna TK_add

S21 - -

- - => S22 (--)
- Otro => Retorna TK_sub

S23 - * => Retorna TK_mul

S23 - /

- * => S25 (Comentario Multilinea)
- / => S28 (Comentario simple)
- Otro => Retorna TK_div

Estados para Comentario

S25 – Comentario multilínea

- [^*] => Permanece en S25
- * => S26

S26 – Fin potencial comentario

- / => S27
- Otro => S25

S27 Fin comentario => Retorna TK_comment

S28 Comentario Simple

- [^\n] => Permanece en S28
- \n => Retorna TK_single_comment

Estados para simbolos

S29 – S37 Reconocimiento de símbolos individuales

- Retorna tokens específicos para cada símbolo

Gramatica libre de contexto

Símbolos Terminales

KW_public, KW_class, KW_static, KW_void, KW_main, KW_String,

KW_args, KW_int, KW_double, KW_char, KW_boolean, KW_true,

KW_false, KW_if, KW_else, KW_for, KW_while, KW_System,

KW_out, KW_println, TK_id, TK_str, TK_char, TK_int, TK_float,

TK_lpar, TK_rpar, TK_lbrc, TK_rbrc, TK_lbrack, TK_rbrack,

TK_semicolon, TK_comma, TK_dot, TK_assign, TK_equal,

TK_notequal, TK_less, TK_greater, TK_lsequal, TK_grtequal,

TK_add, TK_sub, TK_mul, TK_div, TK_inc, TK_dec,

TK_comment, TK_single_comment

Simbolos no Terminales

START, SENTENCIAS, SENTENCIA, DECLARACION, ASIGNACION,

IF, FOR, WHILE, PRINT, TIPO, EXP, EXP2, EXP1, PRIMITIVE

Producciones

Produccion Inicial

START → KW_public KW_class TK_id TK_lbrc

KW_public KW_static KW_void KW_main TK_lpar

KW_String TK_lbrack TK_rbrack KW_args TK_rpar

TK_lbrc SENTENCIAS TK_rbrc TK_rbrc

Produccion Sentencias

SENTENCIAS → SENTENCIA SENTENCIAS | ε

SENTENCIA → DECLARACION | ASIGNACION | IF | FOR | WHILE | PRINT

Produccion Declaraciones

DECLARACION → TIPO TK_id (TK_assign EXP)?

(TK_comma TK_id (TK_assign EXP)?)* TK_semicolon

TIPO → KW_int | KW_double | KW_char | KW_String | KW_boolean

Produccion Estructuras de Control

ASIGNACION → TK_id TK_assign EXP TK_semicolon

IF → KW_if TK_lpar EXP TK_rpar TK_lbrc SENTENCIAS TK_rbrc

(KW_else TK_lbrc SENTENCIAS TK_rbrc)?

FOR → KW_for TK_lpar DECLARACION EXP TK_semicolon

TK_id (TK_inc | TK_dec) TK_rpar TK_lbrc SENTENCIAS TK_rbrc

WHILE → KW_while TK_lpar EXP TK_rpar TK_lbrc SENTENCIAS TK_rbrc

PRINT → KW_System TK_dot KW_out TK_dot KW_println

TK_lpar EXP TK_rpar TK_semicolon

Produccion Expresiones

EXP → EXP2 ((TK_equal | TK_notequal | TK_grtequal |

TK_lsequal | TK_greater | TK_less) EXP2)*

EXP2 → EXP1 ((TK_add | TK_sub) EXP1)*

EXP1 → PRIMITIVE ((TK_mul | TK_div) PRIMITIVE)*

Produccion Primitivos

PRIMITIVE → TK_id | TK_int | TK_float | TK_str | TK_char |

KW_true | KW_false | TK_lpar EXP TK_rpar

START

Define la estructura principal de un programa Java

- Declaración de clase pública
- Método main con firma estándar
- Bloque de instrucciones principales

SENTENCIAS

Secuencia de instrucciones ejecutables
Casos base: Declaraciones, asignaciones, estructuras de control

DECLARACION

Declaración de variables con inicialización opcional
Características

- múltiples variables en una línea separadas por coma
- Inicialización opcional con expresión
- Termina con punto y coma

TIPO

Especificación de tipos de datos Java

- int => int
- double => float
- String => str
- boolean => bool
- char => str

ASIGNACION

Asignación de valores a variables existentes

IF

Estructura condicional con else opcional

FOR

Bucle for con inicialización, condición e incremento
Convertido a while equivalente en Python

- Inicialización: declaración de variable contador
- Condición: expresión booleana
- Incremento: ++ o -- sobre variable
-

WHILE

Bucle while con condición

PRINT

Impresión en consola
Traducción: System.out.println() → print()

EXP (Expresiones Relacionales)

Expresiones con operadores relacionales
Operadores: ==, !=, >=, <=, >, <

EXP2 (Expresiones Aditivas)

Expresiones con suma y resta
Operadores: +, -

EXP1 (Expresiones Multiplicativas)

Expresiones con multiplicación y división
Operadores: *, /

PRIMITIVE

Elementos básicos de expresiones
Tipos:

- Variables (TK_id)
- Literales numéricos (TK_int, TK_float)
- Literales de texto (TK_str, TK_char)
- Booleanos (KW_true, KW_false)
- Expresiones entre paréntesis

Arquitectura del Traductor

Flujo de Procesamiento

1. **Scanner:** Tokenización del código fuente
2. **Parser:** Construcción del AST según la gramática
3. **Generador:** Traducción a Python recorriendo el AST
4. **Entorno:** Gestión de tablas de símbolos y tipos

Clases del AST

- **Expresiones:** AccessVar, Arithmetic, Primitive, Relational
- **Instrucciones:** AssignVar, Block, For, If, InitVar, MainFunc, Print, While
- **Utilidades:** Env (entorno), PythonGenerator (generación de código)